Complete Adiabatic Quantum Search in Unsorted Databases

Nanyang Xu, Xinhua Peng, Mingjun Shi, Jiangfeng Du*
Hefei National Laboratory for Physical Sciences at Microscale and Department of Modern Physics,University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China
*To whom correspondence should be addressed; E-mail: djf@ustc.edu.cn

We propose a new adiabatic algorithm for the unsorted database search problem. This algorithm saves two thirds of qubits than Grover's algorithm in realizations. Meanwhile, we analyze the time complexity of the algorithm by both perturbative method and numerical simulation. The results show it provides a better speedup than the previous adiabatic search algorithm.

PACS numbers: 03.67.Lx, 89.70.-a, 03.65.-w

Quantum computation is a promising way to solve classical hard problems. Several quantum algorithms have been designed to perform classical algorithms with remarkable speedups. The most useful one among these is Grover's algorithm[1] which concerns the problem of searching for a required item in a unsorted database. One common example for this unsorted database search is to find a person's name in a phone book (the items are sorted by names) with only knowing his phone number. Classically, the only way to achieve this is brute-force search[1, 2] which requires an average of $\frac{N}{2}$ quires for N entries in the phone book. However, if the information is stored in a quantum database, to find the right name with Grover's algorithm costs only a time of order $\sqrt{N}$, providing a quadratic speedup.

The main process of Grover's algorithm is, swinging the *index* qubits from an initial uniform state to approach the solution state. The information of the database is not explicitly accessed in the processing of search. Instead, an Oracle is supposed to know all the information in the database and act properly towards a input state depending on whether it denotes the solution[3]. Early experiments[4, 5, 6, 7, 8] of Grover's algorithm constructed this Oracle from a marked state as a functional analog instead of querying the database. For a complete solution of search problem, Kim and coworkers proposed a new approach to realize the Oracle on a quantum database and implemented it in experiment[9]. In this complete approach of Grover's algorithm, extra qubits are used to store the database. Similar method for construction of Grover's Oracle was also theoretically discussed later [2].

While Grover's algorithm is presented in the standard circuit model(*i.e.*, using a sequence of discrete quantum gates), a new model of quantum computation shows up where the state of quantum computer evolves continuously and adiabatically under a certain time-dependent Hamiltonian. This new adiabatic model was soon applied to the database search problem[10] and the original adiabatic search algorithm was proved to have a time complexity of order N, which is the same performance as classical algorithms. More recently, Roland and Cerf[11] recovered the advantage of adiabatic search to order $\sqrt{N}$ (the same with Grover's algorithm), by performing the adiabatic evolution locally. However this adiabatic search algorithm construct the Hamiltonian from a marked state instead of referring to the database, thus it is not a complete search algorithm. And in order to discriminate it from our algorithm, we call it *the marked-state adiabatic search*(MSAS) algorithm in the following paragraphs.

In this letter, we apply the quantum adiabatic computation to unsorted database search problem again and present a new quantum search algorithm. We will put forward a new method to represent the database. By this method the algorithm contains no Oracles and saves $\frac{2}{3}$ of qubits than the complete approach of Grover's algorithm[9]. We also analyze the time complexity by both perturbation method and numerical simulation. The results show it provide a higher speedup than the MSAS algorithm.

As a new quantum computation model, adiabatic algorithm was brought out by Farhi *et al.*[12] and soon became a rapidly expanding field. The idea of this new computation model is to prepare a system in the ground state of a simple initial Hamiltonian, then slowly switch the simple Hamiltonian to a complex Hamiltonian whose ground state encodes the solution to the problem of interest. According to Adiabatic Theorem, the the system stays in the ground state of the instantaneous Hamiltonian if we perform the evolution slowly enough. So finally the state describes the solution to the problem. The time-dependent system Hamiltonian is

$$H(t) = [1 - s(t)]H_i + s(t)H_p, \tag{1}$$

where H_i is the initial Hamiltonian and H_p is the problem Hamiltonian which encodes the solution, and the monotonic function $s(t)$ fulfills $s(0) = 0$ and $s(T) = 1$.

Here let's focus on the unsorted database search problem. To be simplified, the database is a list of (i, v_i) pairs and sorted by i where i denotes *index* and v_i is *value*. Both i and v_i are n-bit binary codes thus the database contained $N = 2^n$ items. The *"unsorted"* property of the database refers to the field *value* not *index*. The unsorted database search problem here is looking for the

corresponding *index* i for a given target *value* t. And we assume that there's only one solution in the database for each search. Next we will describe the process to find the right i which connects to the target t.

The essential part of an adiabatic search algorithm is how to encode the solution in the ground state of problem Hamiltonian. For example, the MSAS algorithm constructs the problem Hamiltonian as $H_p = 1 - |m\rangle\langle m|$ where $|m\rangle$ is exactly the solution state. Thus it is not a complete database search. Obviously for a complete search, the information in database should be represented in quantum forms. Taking the complete approach of Grover's algorithm as an example[2, 9], the database is represented in an operator which satisfies $U_f|i\rangle|0\rangle = |i\rangle|v_i\rangle$. U_f generates the entanglement of qubits to denote the relation between i and v_i, thus both the fields are represented by qubits.

In the present algorithm, however, not both the fields are represented by qubits. We define a database operator as

$$\mathcal{D} = \sum_{i=0}^{N-1} v_i|i\rangle\langle i|. \tag{2}$$

Clearly in this approach, the *index* is represented by qubits while the *value* is store in the strength of interactions. So no extra qubits are needed for the database. The operator $\mathcal{D}$ contains all the information in the database. Thus, we can construct the problem Hamiltonian from $\mathcal{D}$ as

$$H_p = (\mathcal{D} - t)^2, \tag{3}$$

where t is the target *value* which we are looking for.

To test the validity of H_p, we will examine its ground state. To this end, we can write H_p as $\sum_{i=0}^{N-1}(v_i - t)^2|i\rangle\langle i|$. From this form, each diagonal element of H_p is the square of difference between v_i and t. Thus the ground state will be the solution state $|i\rangle$ where v_i equals to t.

Of course this construction provides a valid problem Hamiltonian for the search problem. However, the Hamiltonian in Eq.(3) has a spectral width exponentially growing with the number of qubits, which is hard to realize when the database is large. Thus it is only useful in small-size databases.

To solve this problem, we divide the comparison between v_i and t into n sub-comparisons, each of which is performed for a single bit between them. Thus the database operator should be formed separately for each bit. For the jth bit of *value* we define the bit database operator $\mathcal{D}_j$ as

$$\mathcal{D}_j = \sum_{i=0}^{N-1} v_{ij}|i\rangle\langle i|, \tag{4}$$

where v_{ij} is the jth bit of v_i. Similarly with the operation in Eq.(3), the problem Hamiltonian for each bit is

$$H_p^j = (\mathcal{D}_j - t_j)^2, \tag{5}$$

where t_j is as well the jth bit of t. Consequently, the overall problem Hamiltonian is the summarization of all bit problem Hamiltonians

$$\begin{aligned}\tilde{H}_p &= \sum_{j=0}^{n-1} H_p^j = \sum_{j=0}^{n-1} [\mathcal{D}_j(1 - t_j) + t_j(I - \mathcal{D}_j)] \\ &= \sum_{j=0}^{n-1} (\mathcal{D}_j\bar{t}_j + t_j\bar{\mathcal{D}}_j),\end{aligned} \tag{6}$$

where $\bar{t}_j$ is the complementation of binary bit t_j and $\bar{\mathcal{D}}_j = I - \mathcal{D}_j$.

Also for a test of the validity, we can simplify $\tilde{H}_p$ as

$$\tilde{H}_p = \sum_{i=0}^{N-1} h(v_i, t)|i\rangle\langle i|, \tag{7}$$

where the function $h(v_i, t)$ is the Hamming distance between v_i and t. Thus the state $|i\rangle$ where $h(v_i, t) = 0$ is the ground state of $\tilde{H}_p$ and is also the solution state. Moreover, the spectrum was successfully bounded in a range from 0 to n.

After the preparation of problem Hamiltonian, we will choose an initial Hamiltonian H_i. H_i should be chosen to be noncommutative with H_p to avoid crossing of energy levels[10]. Normally, H_i is

$$H_i = g(\sigma_x^0 + \sigma_x^1 + \cdots + \sigma_x^{n-1}), \tag{8}$$

which means the qubits coupling with a magnetic field at the x-direction and the coupling strength is g. The ground state of H_i is

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} (-1)^{b(j)}|j\rangle, \tag{9}$$

where $b(j)$ is the Hamming distance between j and 0.

In the adiabatic evolution, the system Hamiltonian interpolates from H_i to $\tilde{H}_p$ (*i.e.,* see Eq 1) and the state of the system evolves according to the Schrödinger equation. If this evolution acts adiabatically, the system will always stay on the instantaneous ground state of $H(t)$ and in the end the solution of our problem will show up.

An explicit application is necessary for a clear understanding. Here we perform a 3-bit unsorted database search for example. We randomly generate a database in a list as $\{6, 3, 5, 0, 4, 1, 7, 2\}$. The position of each *value* in the list refers to the *index* which ranges from 0 to 7. For convenience we rewrite the database as binary codes which is {110,011,101,000,100,001,111,010}. Because the database operators and problem Hamiltonian are diagonal, they are expressed by only the diagonal elements.

$$\begin{aligned}\mathcal{D}_0 &= diag\{0, 1, 1, 0, 0, 1, 1, 0\} \\ \mathcal{D}_1 &= diag\{1, 1, 0, 0, 0, 0, 1, 1\} \\ \mathcal{D}_2 &= diag\{1, 0, 1, 0, 1, 0, 1, 0\}.\end{aligned} \tag{10}$$

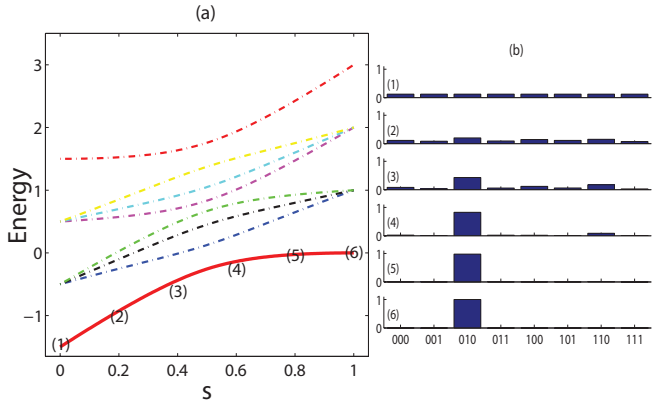

FIG. 1: Process of the adiabatic evolution to search for *value* 5 in the mentioned database. (a)The instantaneous eigenvalues of the system Hamiltonian as a function of s. The solid line represented the energy level of ground state. (b)Occupation probabilities of the system for the computational basis during the adiabatic evolution in the numerical simulation. The system starts from a uniform state and evolves to the solution state $|010\rangle$ which shows *value* 5 is on *index* 2. The parameter g of H_i in this example is 0.5.

After building the quantum database, the problem Hamiltonian can be constructed for each search task. For example, if we want to find the position of the *value* 5 which is 101 in binary, the problem Hamiltonian is

$$\begin{aligned}\tilde{H}_p &= \bar{\mathcal{D}}_0 + \mathcal{D}_1 + \bar{\mathcal{D}}_2 \\ &= diag\{2,2,0,2,1,1,1,3\}. \end{aligned} \quad (11)$$

To perform the adiabatic evolution, we initially prepare the system on the state of Eq.(9), and adiabatically switch the system Hamiltonian from a initial Hamiltonian in Eq.(8) to the problem Hamiltonian in Eq.(11). Finally the system will on the ground state of $\tilde{H}_p$ which is the state $|2\rangle$. After measurement, we can get the knowledge that *value* 5 is on position 2. Fig.1 shows the process of the adiabatic evolution for this search.

For the practical usefulness of an algorithm, the occupied amount of resources is an important aspect. Without doubts, the number of qubits needed for our algorithm equals to the bit width of the *index* because only the *index* field is represented by qubits. Thus the spatial complexity is n. As a comparison, since both the fields *value* and *target* are represented by qubits, the spatial complexity of the complete approach of Grover's algorithm[9] is $3n$. Although the MSAS algorithm also have a spatial complexity of n, it is not a complete database search. Thus our algorithm has the best spatial complexity in the quantum algorithms for complete database searches.

To evaluate the time complexity of our algorithm, a decisive mathematical analysis is not possible. Therefore in this letter, we use both the perturbative method[13] and numerical simulation[12] to examine the situation of the time complexity.

In the perturbative approach[13], the time cost of a adiabatic algorithm by either *global* or *local* evolution can be written as

$$\begin{aligned} T_{local} &\propto \sqrt{T_{global}} \propto \sqrt{|S^-|/|S^+|}, \quad (12) \\ S^+ &\approx \{z : h(z,f) < m_c\}, m_c \propto \frac{\log 1/\delta}{\log \zeta^+} \\ S^- &\approx \{z : E_z < E_c\}, E_c \propto \frac{\log 1/\zeta^-}{\log 1/\delta}, \quad (13) \end{aligned}$$

where S^+ is a set containing the eigenstates of the problem Hamiltonian which have a small Hamming distance towards the solution state $|f\rangle$, while S^- contains the ones which have low energy levels. $|S|$ is the cardinality of set S. $\zeta^\pm$ are dimensionless parameters which are defined as $\zeta^\pm \equiv \zeta(s^* \pm \epsilon_0)$. Here $\zeta(t) = \frac{s(t)}{1-s(t)}$ and s^* is the position of the minimum gap between the ground and first exited state. ϵ_0 and δ are small numbers.

To apply this result to our algorithm, we assume that the minimum gap is on the central position of s, thus we can get $\zeta^+ = 1/\zeta^- > 1$. Then we define a small number $\Omega \equiv \frac{\log 1/\delta}{\log \zeta^+}$. Because the degeneracy of energy levels in the problem Hamiltonian in Eq.(7) is C_n^i where i is the ith energy level, Eq.(13) goes as

$$\begin{aligned} |S^+| &\approx \sum_{i=0}^{i<m_c} C_n^i, m_c \propto \Omega \\ |S^-| &\approx \sum_{i=0}^{i<E_c} C_n^i, E_c \propto 1/\Omega. \end{aligned} \quad (14)$$

Here, Ω is a small number and is not supposed to increase with n, so only some low energy levels will be in S^- and $|S^+|$ is a comparatively small positive integer. Since $\sum_{i=0}^{n-1} C_n^i = N$, for the worst case, we can take Eq.(12) as $T_{global} \propto |S^-| \propto N^\alpha$ where $\alpha < 1$ is a constant.

To derive a more accurate range for α, we performed a numerical simulation[12] for randomly generated databases with the bit width of *index* sized from 5 to 16. For each bit, we randomly generated 50 instances of database search. Then we performed a numerical global evolution using four-order self-adapted Runge-Kutta method to get a success probability of range $[0.12, 0.13]$ for each instance. The mean time for each bit is shown in Fig.2. By fitting the mean time, we obtain $\alpha = 0.81$. For a comparison, we simulated the time complexity of MSAS algorithm using the same environment. The value α of MSAS algorithm obtained from the fitting is 1.02. The result of simulation fit well with theoretical expectation where α is 1[10].

In Fig.2, the running time of our algorithm grows much more slowly than the MSAS algorithm. This result

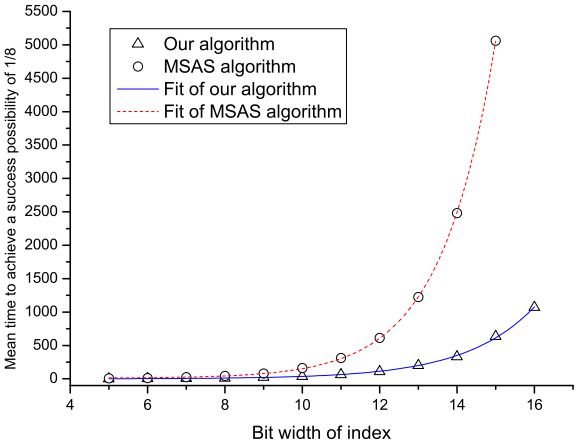

FIG. 2: Comparison of running time between our algorithm and MSAS algorithm to achieve a success probability of 1/8 as the function of bit width of *index*. The solid line is the fit of the triangles each of which represent the mean simulation result of 50 instances (32 instances for $n = 5$) of our algorithm, while the dash line is the fit of the circles each of which is that of the MSAS algorithm. The error range of fitting the triangles is $\pm 2\%$ and that for the circles is $\pm 0.3\%$

matches well with the expectation from the perturbative analysis. Both the results show that our algorithm has a better performance in time complexity than the MSAS algorithm. And because local evolution can provide a quadratic speedup over global evolution, theoretically the time complexity of our algorithm by local evolution can be reduced to less than order $\sqrt{N}$, even lower than than the complexity of Grover's algorithm.

To be concluded, we introduce a new algorithm for quantum search problem by adiabatic evolution. We use another method to represent the quantum database in this algorithm and it saves $\frac{2}{3}$ of qubits than the complete approach of Grover's algorithm[9]. We use both the emerging perturbative method of adiabatic algorithm and numerical simulation to analyze the time complexity in this algorithm. The results show that it provides a higher speedup than the MSAS algorithm and potentially has a better performance than Grover's algorithm. This algorithm can be experimentally verified in NMR or ion-trap systems[14, 15].

The authors thank Zeyang Liao, Dieter Suter and Guilu Long for discussions and comments. This work was supported by National Nature Science Foundation of China, the CAS, Ministry of Education of PRC.

[1] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).
[2] Y. L. Ju, I. M. Tsai, and S. Y. Kuo, Ieee Transactions on Circuits and Systems I-Regular Papers **54**, 2552 (2007).
[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge Univ. Press, Cambridge, 2000).
[4] I. L. Chuang, N. Gershenfeld, and M. Kubinec, Physical Review Letters **80**, 3408 (1998).
[5] J. L. Dodd, T. C. Ralph, and G. J. Milburn, Physical Review A (Atomic, Molecular, and Optical Physics) **68**, 42328 (2003).
[6] L. M. K. Vandersypen, M. Steffen, M. H. Sherwood, C. S. Yannoni, G. Breyta, et al., Applied Physics Letters **76**, 646 (2000).
[7] K. A. Brickman, P. C. Haljan, P. J. Lee, M. Acton, L. Deslauriers, et al., Physical Review A (Atomic, Molecular, and Optical Physics) **72**, 50306 (2005).
[8] G. Long, H. Yan, Y. Li, C. Tu, J. Tao, et al., Physics Letters A **286**, 121 (2001).
[9] K. Jaehyun, L. Jae-Seung, and L. Soonchil, Physical Review A (Atomic, Molecular, and Optical Physics) **65**, 54301 (2002).
[10] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, quant-ph **0001106** (2000).
[11] J. Roland and N. J. Cerf, Physical Review A **65**, 6 (2002).
[12] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, et al., Science **292**, 472 (2001).
[13] M. H. S. Amin, Phys Rev Lett **100**, 130503 (2008).
[14] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, et al., quant-ph **08081935** (2008).
[15] A. Friedenauer, H. Schmitz, J. T. Glueckert, D. Porras, and T. Schaetz, Nature Phys **4**, 757 (2008).